

Référence : BDATPDIRECCTEECF-Ete2023_346813_20221209163721

A. Spécifications techniques

Serveur locale :

Serveur de base de données

- Version du serveur : 10.4.27-MariaDB - mariadb.org binary distribution

Serveur Web

- Version de PHP : 8.2.0

PhpMyAdmin

- Version : 5.2.0

Serveur en ligne :

- Heroku

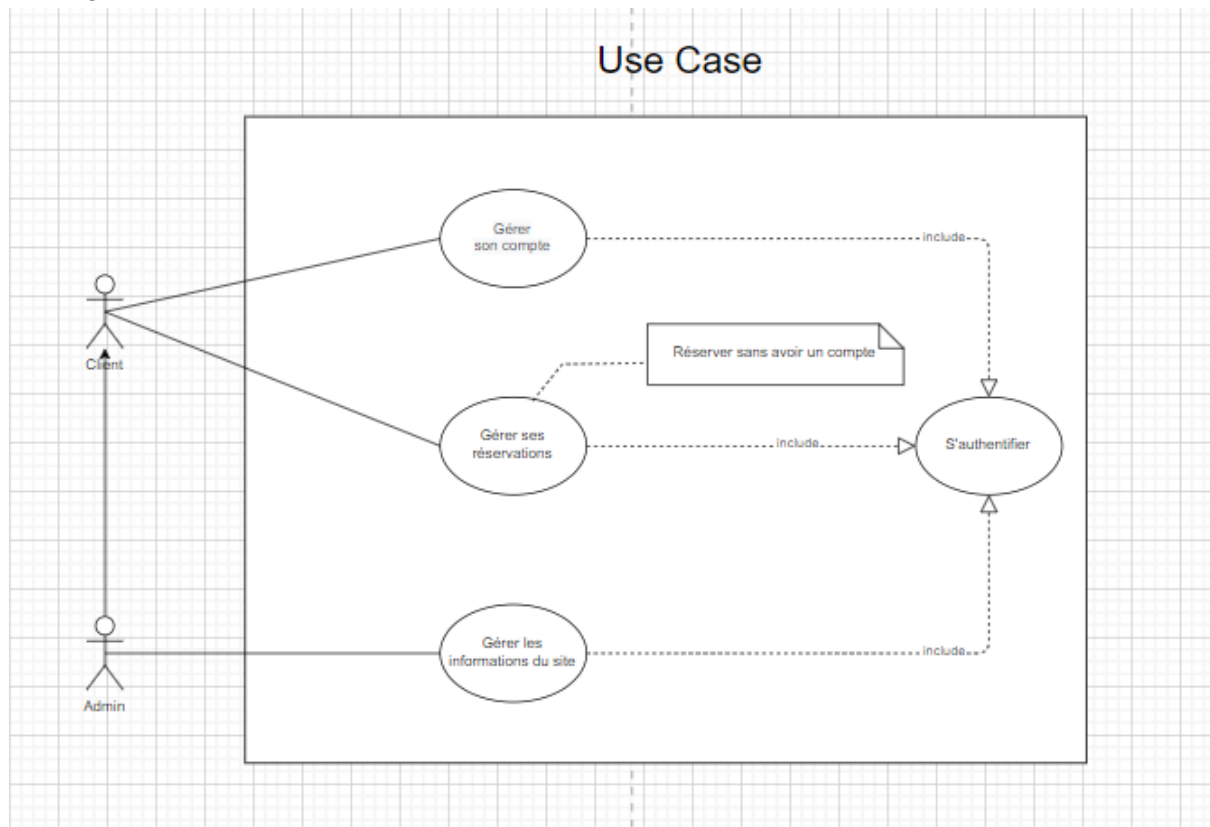
Pour le front :

- HTML 5
- CSS 3
- Bootstrap

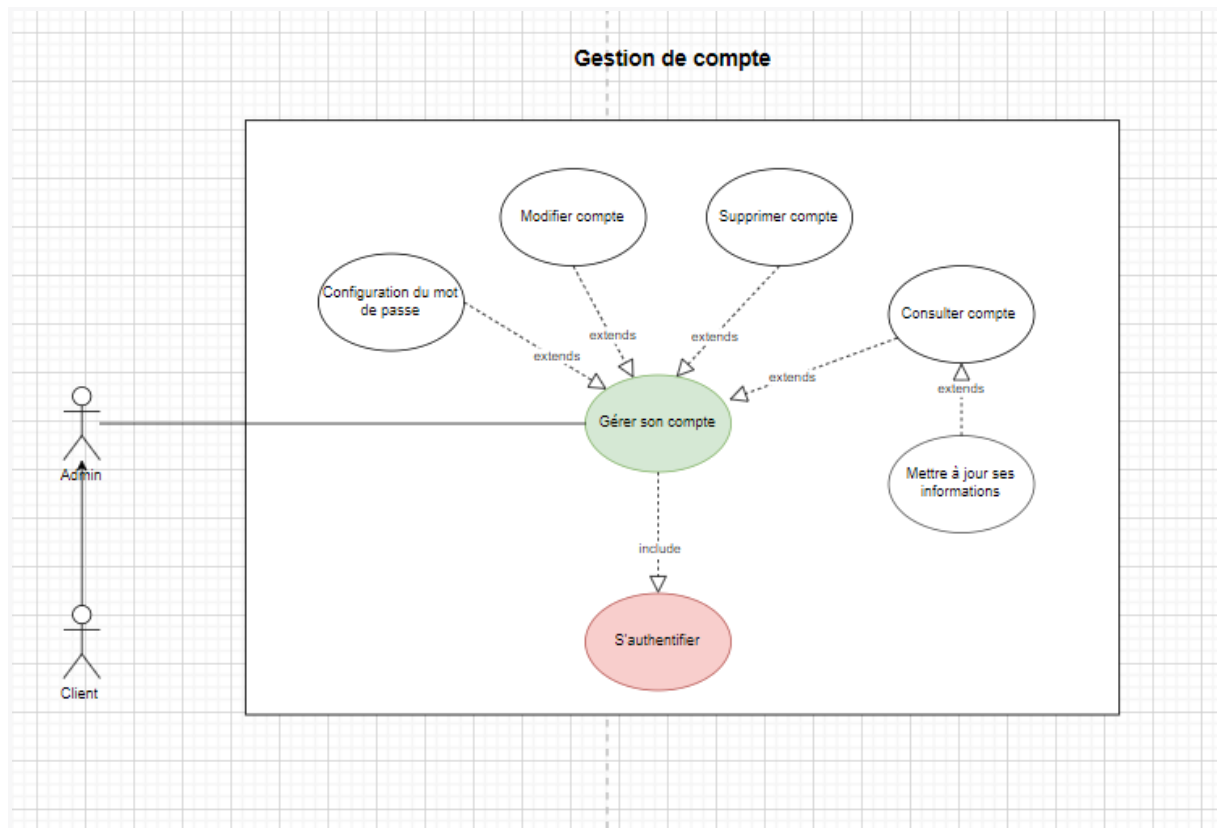
Pour le back :

- PHP 8.2.0 sous PDO
- Symfony 6.2.6
- MySQL

B. Diagramme de Cas d'utilisation



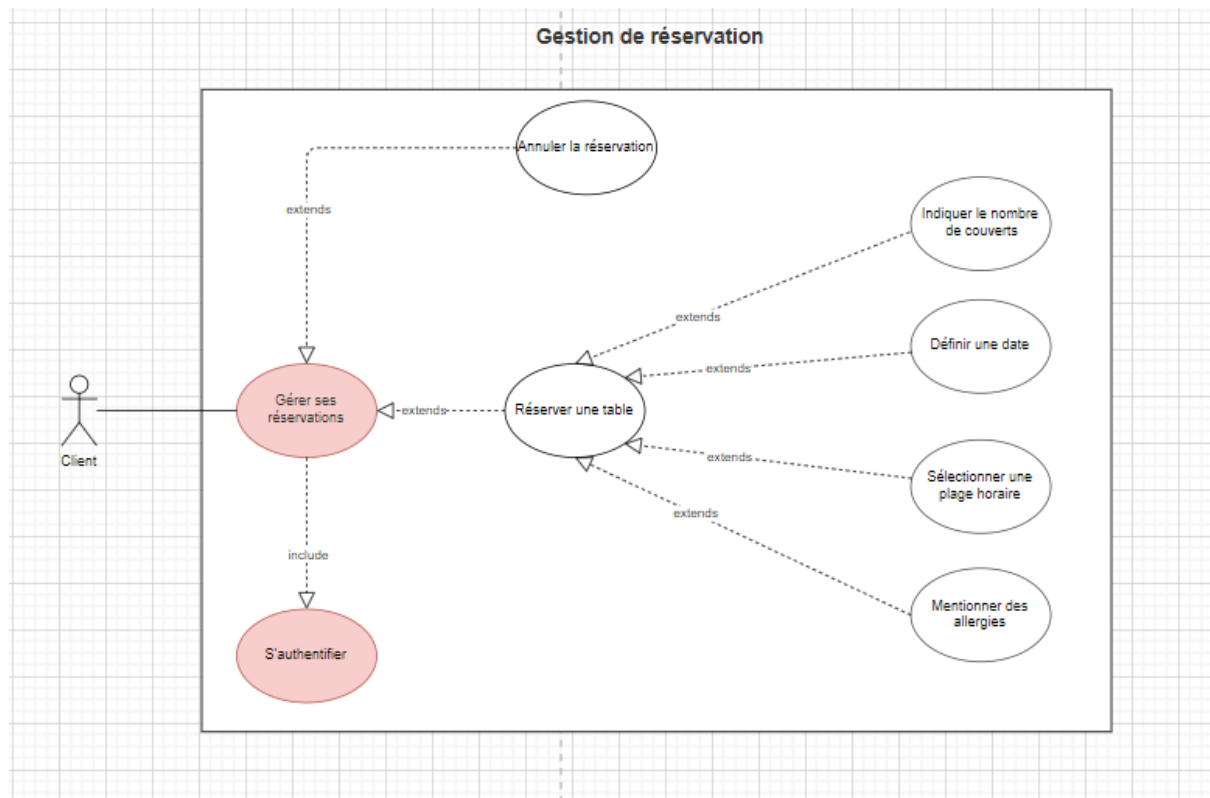
Le Use Case que nous étudions implique deux acteurs principaux : un administrateur et un client (un utilisateur). Les deux acteurs ont la possibilité de s'authentifier via une connexion. L'administrateur a la possibilité de gérer toutes les informations du site, tandis que le client peut gérer son compte et ses réservations sans avoir l'obligation de se créer un compte.



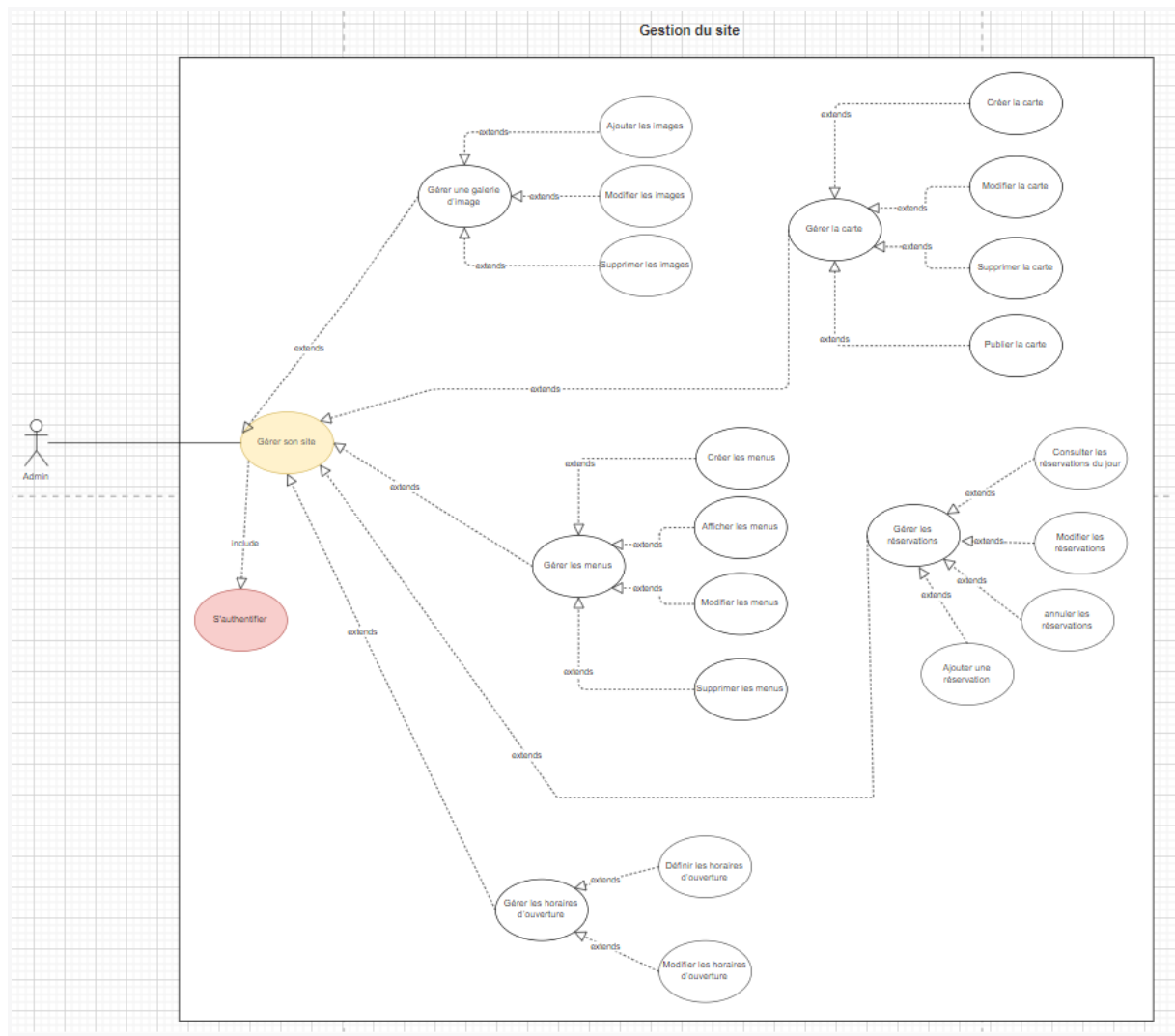
Dans ce Use Case, l'administrateur et le client pourront accéder à différentes fonctionnalités de gestion de compte, telles que la consultation de leur profil, la mise à jour de leurs informations personnelles, ou encore la suppression de leur compte. Ils devront s'authentifier à la plateforme avant d'accéder à ces actions.

En outre, l'administrateur aura des privilèges supplémentaires, comme la gestion des comptes clients, la création de nouveaux comptes, ou encore la modification des informations de l'ensemble des comptes.

Le client, quant à lui, pourra uniquement gérer son propre compte et les réservations associées. Il pourra consulter les détails de ses réservations, les modifier ou les annuler.



Dans ce Use Case, nous allons aborder la gestion des réservations qui implique l'acteur client et administrateur. Ce dernier pourra réserver une table en se connectant ou non à la plateforme. Il devra alors remplir des champs obligatoires tels que la date de réservation, le créneau horaire qui lui convient, le nombre de personnes et la présence ou non d'allergies. Enfin, le client aura la possibilité d'annuler sa réservation si besoin.

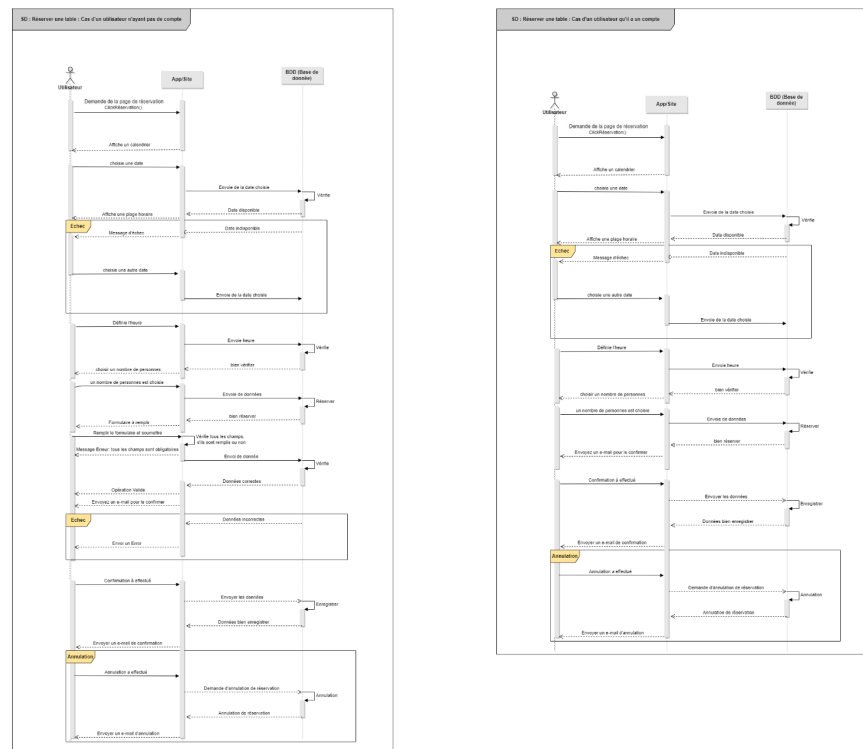


Dans ce Use Case de la gestion du site, l'administrateur est l'acteur principal qui peut gérer plusieurs fonctionnalités. Pour cela, il doit s'authentifier à la plateforme. Les fonctionnalités qu'il peut gérer sont :

- La galerie d'images : il peut ajouter, modifier et supprimer des images.
- La carte : il peut créer, modifier, supprimer et publier la carte.
- Les menus : il peut créer, afficher, modifier et supprimer les menus.
- Les réservations : il peut consulter, modifier, annuler et ajouter une réservation.
- Les horaires d'ouverture : il peut définir et modifier les horaires d'ouverture.

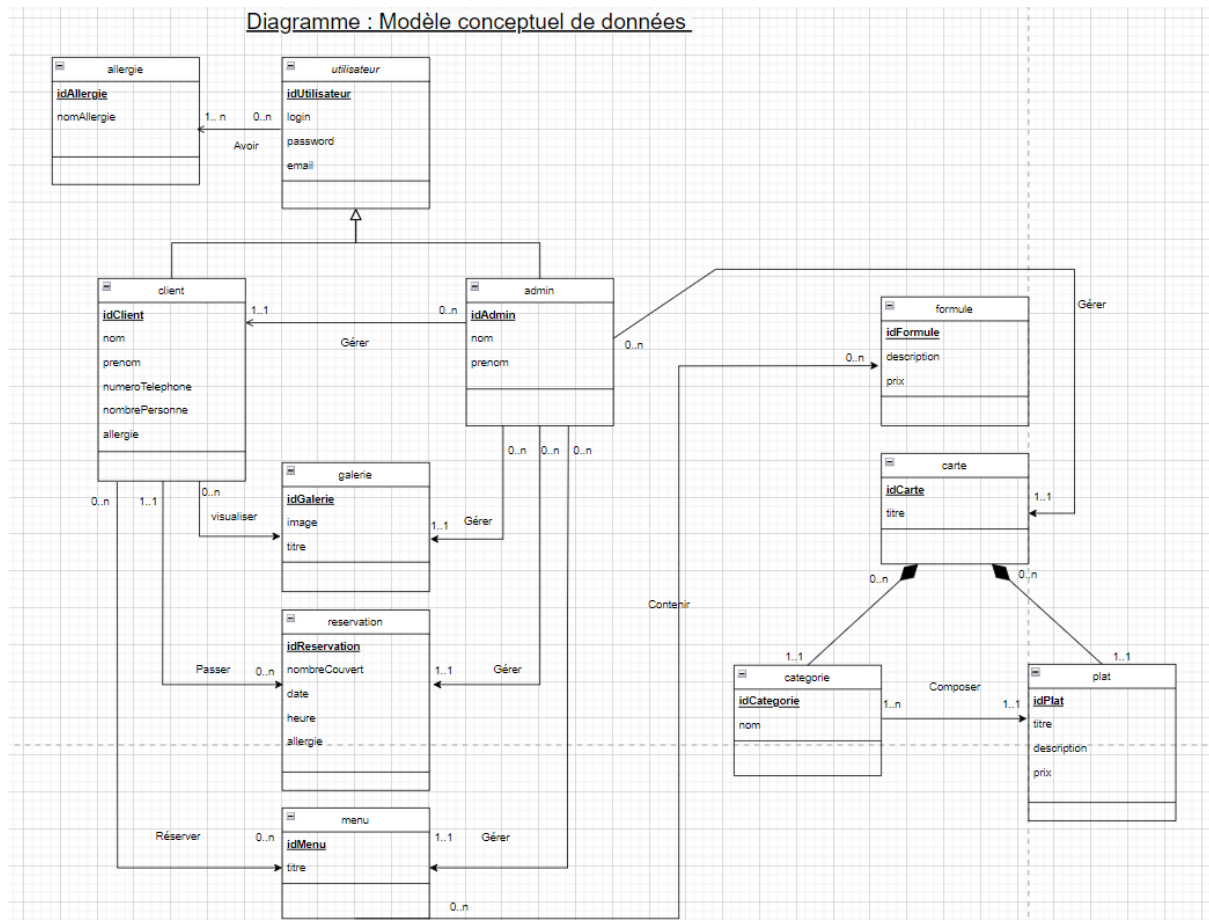
C. Diagrammes de séquence

Diagramme de séquence



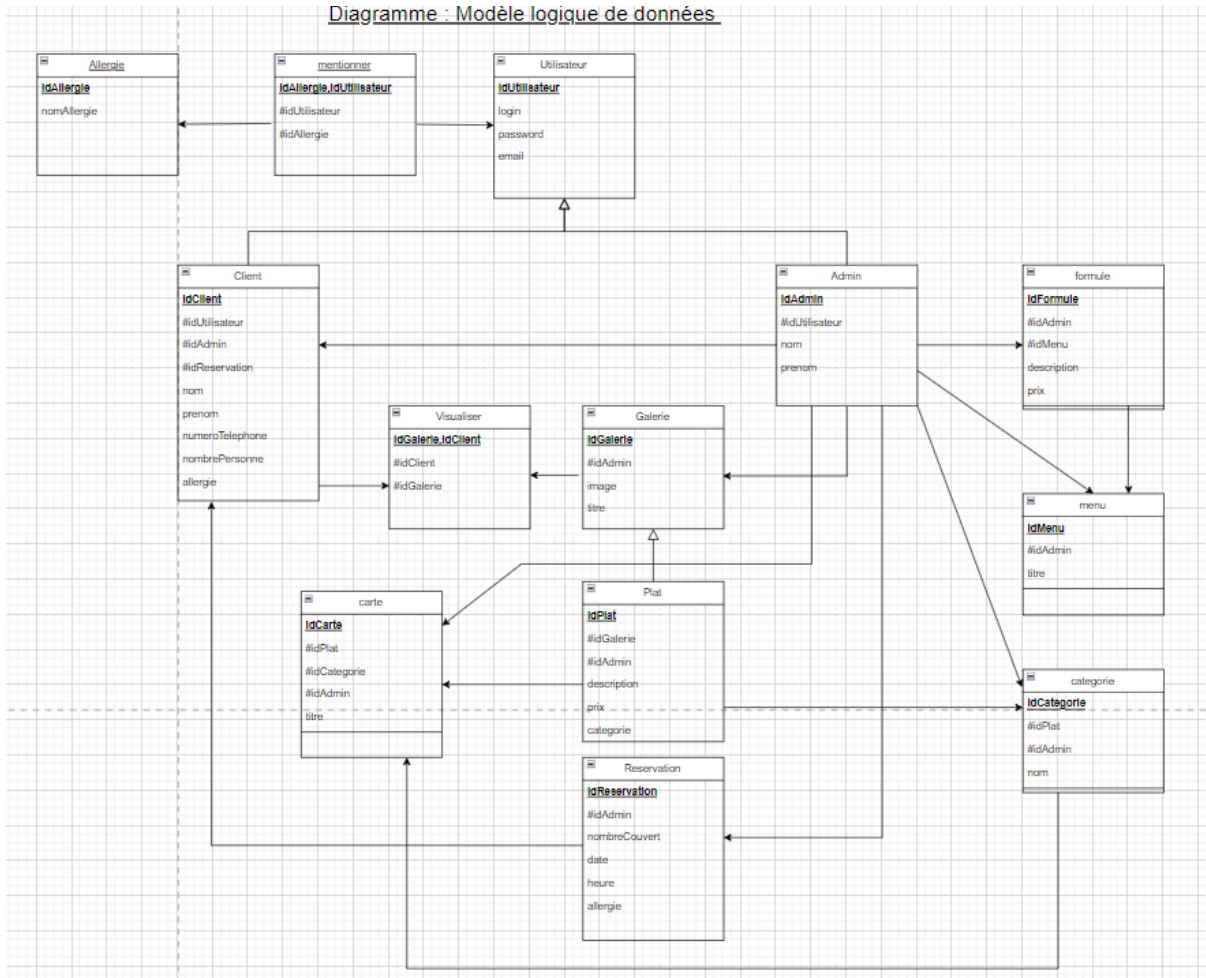
Dans ce diagramme de séquence pour la fonctionnalité "Réserver une table", on a ici par exemple le cas où l'utilisateur n'a pas de compte. Il peut directement cliquer sur le bouton "Réserver" sur le site et accéder à la page de réservation. Une fois qu'il a choisi une date dans le formulaire de réservation, le système vérifie auprès de la base de données si cette date est disponible. Si la date n'est plus disponible, l'utilisateur reçoit un message d'erreur et doit choisir une autre date. Si la date est disponible, l'utilisateur peut sélectionner un créneau horaire et fournir d'autres informations telles que le nombre de personnes et les allergies. Si toutes les informations sont correctes, l'utilisateur reçoit un e-mail de confirmation de sa réservation. S'il souhaite annuler sa réservation, il peut également recevoir un e-mail de confirmation en conséquence.

D. Diagramme de classe (ou Méthode MERISE)



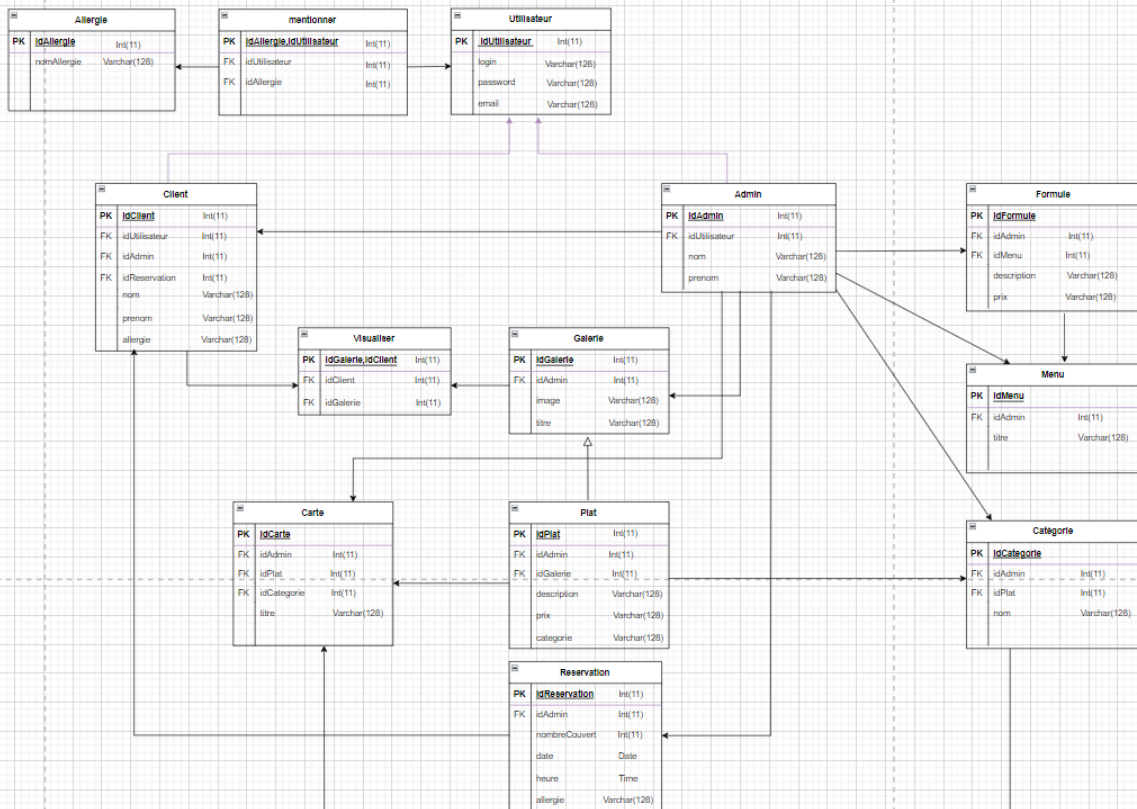
Le MCD est le modèle conceptuel de données, qui est une représentation claire des données du système d'information à concevoir. Dans ce dernier, on définit les tables nécessaires dans la base de données ainsi que leur relation. Par exemple, la table "réservation" contient cinq propriétés : l'ID, le nombre de personnes, la date, l'heure et les allergies. Cette table a deux relations : une avec la table "client", où le client peut passer 0 ou plusieurs réservations et dans l'autre sens une réservation appartient à un seul client ; et une relation avec la table "administrateur", où l'administrateur peut gérer 0 ou plusieurs réservations et dans l'autre sens une réservation peut être gérée par un seul administrateur.

Diagramme : Modèle logique de données



Dans cette section, nous allons examiner le MLD, qui est le modèle logique de données. Avant cela, nous avons étudié le MCD, qui représente les données du système d'information à concevoir. Le MLD supprime toutes les relations et actions entre les tables et injecte les ID des tables entre eux selon les relations définies dans le MCD. Les clés primaires sont les premiers ID dans les tables, tandis que les clés étrangères sont les ID injectés après. Par exemple, l'ID de l'administrateur est injecté dans la table client en tant que clé étrangère pour indiquer qu'un administrateur peut gérer 0 ou plusieurs clients.

Diagramme : Modèle physique de données



Le MPD, qui signifie modèle physique de données, est un modèle qui intègre les types de propriétés et les clés étrangères définies dans le MLD. Dans ce modèle, chaque propriété est associée à un type, tels que "integer", "varchar", "date", "heure", etc. Les identifiants (id) sont tous définis comme des entiers (int). Par exemple, dans la table admin, l'id est de type int, la clé étrangère idUtilisateur est également de type int, et les propriétés nom et prénom sont de type Varchar. Les clés étrangères sont représentées par la notation FK, qui signifie foreign key.